CLAIMS

1. A method of communicating data, via a device driver, between an application and an interface having at least one feature to which a corresponding interface identifier is assigned, the assignment of the interface identifier to the feature being susceptible to change after at least one event, the method comprising:

for at least one said feature, storing a corresponding logical identifier;

providing the logical identifier to the application for directing communication associated with the corresponding feature between the device driver and the application; and

maintaining correspondence between the or each logical identifier and the or each feature independently of the interface identifier assigned to the or each feature so that communication between the application and the device driver directed using a given logical identifier remains associated with the corresponding given feature following a change in the assignment of the corresponding interface identifier to the feature.

2. A method according to Claim 1, wherein communication between the interface and the device driver is directed based on the or each interface identifier.

3. A method according to ~~any preceding claim~~ Claim 1, including compiling a list of logical identifiers and corresponding interface identifiers for all features meeting pre-determined criteria.

4. A method according to ~~any preceding claim~~ Claim 1, wherein the device driver is arranged to communicate the interface identifier assigned to a logical identifier to the application on request.

5. A method according to ~~any preceding claim~~ Claim 1, wherein the device driver is arranged to accept requests from an application to define connections between physical devices connected to the bus using at least one logical identifier in place of an

interface identifier.

Claim 1

6. A method according to ~~any preceding claim~~ wherein the application is arranged to communicate with the device driver via device manager means.

Claim 1

7. A method according to ~~any preceding claim~~ wherein at least one said feature of the interface comprises a peripheral connected to the interface and the corresponding interface identifier comprises the physical address assigned to that peripheral, the logical identifier comprising a logical address assigned to the peripheral.

8. A method according to Claim 7, wherein maintaining correspondence includes interrogating each peripheral to which a logical address is assigned to determine the physical address assigned to the peripheral following a bus reset.

9. A method according to Claim 4 and Claim 7 or Claim 8, wherein communicating the interface identifier for a given peripheral comprises communicating the physical address of the peripheral and also includes communicating a unique node identifier containing further information identifying the peripheral.

Claim 1

10. A method according to ~~any preceding claim~~, wherein at least one said feature of the interface comprises a channel of defined parameters available via the interface and the corresponding interface identifier comprises the interface channel number, the logical identifier comprising a logical channel identifier.

11. A method according to Claim 10, wherein the device driver is arranged to receive a request from an application to allocate a channel of defined parameters and to return a logical channel identifier if allocation is successful.

12. A method according to Claim 10 ~~or 11~~, wherein the device driver is arranged to accept a preferred interface channel number and to allocate the preferred interface channel if available, and to allocate a free channel if the preferred interface channel

is not available or if no preferred interface channel is specified.

13. A method according to Claim 10, ~~11 or 12,~~ wherein the device driver is arranged to receive an identifier of a preferred interface channel, to recognise a pre-determined key in place of a valid interface channel number as indicating that no preferred interface channel is specified, and to report an error to the application if other invalid interface channel numbers are specified.

14. A method according to Claim 10, ~~11, 12 or 13~~ wherein the device driver is arranged to communicate the interface channel number to the application, and at least one other parameter selected from: the maximum rate allocated to the channel; the rate currently available; the number of connections (if any) using the channel; and the identifiers of each connection using the channel.

15. A method according to ~~any preceding claim~~ Claim 1 wherein the device driver is arranged to accept requests from an application to define one or more connections between physical devices attached to the interface by reference to logical addresses and logical channel identifiers.

16. A method according to ~~any preceding claim~~ Claim 1 wherein the device driver is arranged to establish at least a broadcast connection.

17. A method according to ~~any preceding claim~~ Claim 1 wherein the device driver is arranged to signal one or more events to an application, the events preferably including reset of the bus (preferably beginning and end of reset) and a change in bus topology or channel or connection parameters.

18. A device driver for effecting communication between an application and an interface having at least one feature to which an interface identifier is assigned, the or each interface identifier being liable to change after at least one event, the device driver comprising:

means for storing at least one logical identifier corresponding to a respective

interface identifier;

means for providing the logical identifier to the application for directing communication associated with the corresponding feature between the device driver and the application; and

means for maintaining correspondence between the or each logical identifier and the or each feature independently of the interface identifier assigned to the or each feature so that communication between the application and the device driver directed using a given logical identifier can remain associated with the corresponding given feature following a change in the assignment of the corresponding interface identifier to the feature.

19. A device driver according to Claim 18, wherein the device driver is implemented in software, preferably executable by processing means which runs the or each application.

Claim 18

20. A device driver according ~~to any of Claims 18 to 19~~, wherein the device driver is arranged to compile a list of logical identifiers and corresponding interface identifiers for all features meeting pre-determined criteria.

Claim 18

21. A device driver according to ~~any of Claims 18 to 20~~ including means for communicating the interface identifier assigned to a logical identifier to the application on request.

Claim 18

22. A device driver according to ~~any of Claims 18 to 21~~, including means for accepting a request from an application to define connections between physical devices connected to the bus using at least one logical identifier in place of an interface identifier.

Claim 18

23. A device driver according to ~~any of Claims 18 to 22~~, wherein at least one said feature of the interface comprises a peripheral connected to the interface and the corresponding interface identifier comprises the physical address assigned to that peripheral, the logical identifier comprising a logical address assigned to the

peripheral.

24. A device driver according to Claim 23, arranged to interrogate each peripheral to which a logical address is assigned to determine the physical address assigned to the peripheral following a bus reset.

25. A device driver according to ~~Claim~~ 21 and Claim 23 or 24, wherein the means for communicating the interface identifier for a given peripheral comprises means for communicating the physical address of the peripheral and also includes means for communicating a unique node identifier containing further information identifying the peripheral.

26. A device driver according to ~~any of Claims 18 to 25~~ Claim 18, wherein at least one said feature of the interface comprises a channel of defined parameters available via the interface and the corresponding interface identifier comprises the interface channel number, the logical identifier comprising a logical channel identifier.

27. A device driver according to Claim 26 including channel allocating means arranged to receive a request from an application to allocate a channel of defined parameters and to return a logical channel identifier if allocation is successful.

28. A device driver according to Claim 27, wherein the channel allocating means is arranged to accept a preferred interface channel number and to allocate the preferred interface channel if available, and to allocate a free channel if the preferred interface channel is not available or if no preferred interface channel is specified.

29. A device driver according to Claim 27 ~~or Claim 28~~, wherein the channel allocating means is arranged to receive an identifier of a preferred interface channel, to recognise a pre-determined key in place of a valid interface channel number as indicating that no preferred interface channel is specified, and to report an error to the application if other invalid interface channel numbers are specified.

30. A device driver according to Claim 26, ~~27, 28 or 29~~ including means for communicating the interface channel number to the application, and at least one other parameter selected from: the maximum rate allocated to the channel; the rate currently available; the number of connections (if any) using the channel; and the identifiers of each connection using the channel.

Claim 18

31. A device driver according to ~~any of Claims 18 to 30~~ including means arranged to accept requests from an application to define one or more connections between physical devices attached to the interface by reference to logical channel identifiers and, in the case of a request to define a point to point connection, by reference to logical addresses of the peripherals.

Claim 18

32. A device driver according to ~~any of Claims 18 to 31~~, including means arranged to establish at least a broadcast connection on request by an application.

Claim 18

33. A device driver according to any of ~~Claims 18 to 31~~, including means for signalling one or more events to an application, the events preferably including reset of the bus (preferably beginning and end of reset) and a change in bus topology or channel or connection parameters.

34. A data processing system comprising:

run-time engine means for running an application;

interface means for connection to at least one device, the interface having at least one feature to which an interface identifier is assigned, the or each interface identifier being liable to change after at least one event; and

device driver means according to any of Claims 18 to 33.

35. A data processing system according to Claim 34 implemented in a receiver/decoder which includes means for receiving broadcast data, the interface being arranged for connection to a digital video recorder or digital display device or computer for display or storage of at least a portion of the received data.

36. A receiver/decoder according to Claim 35, wherein the device driver means is arranged to cooperate with further device driver means for modifying the received data stream to produce a modified data stream for passing to said interface.

37. A receiver/decoder according to Claim 35 ~~or 36~~, wherein the interface conforms to the IEEE 1394 standard or a variant thereof.

38. A receiver/decoder according to Claim 35, ~~36 or 37~~, wherein the application is run in an interpreted language and the device driver is compiled.

39. A receiver/decoder according to Claim 35, ~~36, 37 or 38~~, wherein the device driver is arranged to transmit commands for controlling a digital video recorder from the application and/or to receive data concerning the information stored on the digital video recorder.

40. A receiver/decoder according to Claim 39, wherein the data to be communicated includes data in MPEG format.

41. A device driver for use in a receiver/decoder having run-time-engine means for running an application and an IEEE 1394 interface to which at least one peripheral can be connected, the or each peripheral capable of having a respective physical address assigned thereto, the interface being capable of providing at least one communication channel, the or each channel having a respective real channel identifier assigned thereto, the real channel identifier assigned to each channel and the address assigned to each peripheral being liable to change after a bus reset, the device driver being arranged for facilitating communication between the application and the IEEE 1394 interface, the device driver comprising:

means for storing at least one logical address corresponding to a respective peripheral and for storing at least one logical channel identifier corresponding to a respective real channel;

means for providing the logical address to the application for directing communication between the device driver and the application;

channel allocating means for receiving a request from an application to allocate a communication channel, and, if a suitable communication channel is available, for allocating the available suitable communication channel and providing a logical channel identifier to the application for directing communication between the device driver and the application;

connection allocating means for receiving a request from an application to allocate a connection between peripherals attached to the interface using a channel identified by said logical channel identifier, and allocating a connection if possible, wherein, in the case of a request for a point–to–point connection between peripherals, the peripherals are identified using said logical addresses;

peripheral identity means arranged to receive a request from an application to identify a peripheral corresponding to a given logical address and, in response thereto, to communicate the physical address of the corresponding peripheral and to communicate a unique node identifier containing further information identifying the peripheral;

event signalling means for signalling to the application one of a plurality of events including an interface bus reset; and

channel identity means arranged to receive a request from an application to identify a channel corresponding to a given logical channel identifier and, in response thereto, to communicate the interface channel identifer of the corresponding channel and to communicate at least one further parameter of the channel indicating at least one of maximum allocated channel bandwidth and currently available channel bandwidth;

wherein the channel allocating means is arranged to receive an identifier of a preferred real channel and to allocate the preferred real channel if available, and to allocate a free channel if the preferred real channel is not available or if the preferred real channel identifier comprises a pre–determined key in place of a valid real channel identifier and to report an error to the application if the preferred channel identifier corresponds to an invalid real channel identifier other than the pre–determined key.

42. A receiver/decoder comprising means for receiving broadcast data; run–time–engine means for running at least one application; IEEE 1394 interface means for

connection to at least one peripheral device; and device driver means according to Claim 41 for interfacing the or each application to the IEEE 1394 interface means, and means for transporting received data to the IEEE 1394 interface.

43. A method of communicating data substantially as herein described with reference to and as illustrated in the accompanying drawings.

44. A device driver substantially as herein described with reference to the accompanying drawings.

45. A data processing system substantially as herein described with reference to the accompanying drawings.

46. A receiver/decoder substantially as herein described with reference to the accompanying drawings.